



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1988

Analysis of existing advanced data models
and their applicability as a model for a
multimedia database management system.

Enbody, Diane M.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/22886>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

ODDIE V. ... LIBRARY
NAVAL V. ... SCHOOL
MONTEREY, CALIFORNIA 95943-5002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

E43655

**ANALYSIS OF EXISTING ADVANCED DATA
MODELS AND THEIR APPLICABILITY
AS A MODEL FOR A MULTIMEDIA
DATABASE MANAGEMENT SYSTEM**

by

Diane M. Enbody

December 1988

Thesis Advisor:

C. Thomas Wu

Approved for public release; distribution is unlimited.

T241905

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution Availability of Report		
2b Declassification/Downgrading Schedule			Approved for public release; distribution is unlimited.		
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)		
6a Name of Performing Organization		6b Office Symbol	7a Name of Monitoring Organization		
Naval Postgraduate School		(If Applicable) 52	Naval Postgraduate School		
6c Address (city, state, and ZIP code)			7b Address (city, state, and ZIP code)		
Monterey, CA 93943-5000			Monterey, CA 93943-5000		
8a Name of Funding/Sponsoring Organization		8b Office Symbol	9 Procurement Instrument Identification Number		
		(If Applicable)			
8c Address (city, state, and ZIP code)			10 Source of Funding Numbers		
			Program Element Number	Project No	Task No
					Work Unit Accession No
11 Title (Include Security Classification) Analysis of Existing Advanced Data Models and Their Applicability as a Model for a Multimedia Database Management System					
12 Personal Author(s) Diane M. Enbody					
13a Type of Report		13b Time Covered		14 Date of Report (year, month, day)	
Master's Thesis		From To		December 1988	
15 Page Count					
83					
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	Multimedia Database Management System, Data Model, Multimedia		
19 Abstract (continue on reverse if necessary and identify by block number)					
<p>The constant and fast-paced changes that are taking place in computer technology have brought forth a vast array of new applications. It is now possible to store not only standard alphanumeric data but also graphical, voice, and sound as well. This has opened up enormous possibilities for expanding the use of these data forms. This thesis is directed at exploring those possibilities and several current research projects that are attempting to model a multimedia database system. These models will be explored in terms of both their strong points and their weak points. Two possible applications will then be looked at in terms of how they could be modeled using each of these three models.</p>					
20 Distribution/Availability of Abstract			21 Abstract Security Classification		
<input checked="" type="checkbox"/> unclassified/unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users			Unclassified		
22a Name of Responsible Individual			22b Telephone (Include Area code)		22c Office Symbol
Professor C. Thomas Wu			(408) 646-3391		52Wq

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted

security classification of this page

All other editions are obsolete

Unclassified

Approved for public release; distribution is unlimited.

**Analysis of Existing Advanced Data Models
and Their Applicability as a Model
for a Multimedia Database Management System**

by

Diane M. Enbody
Lieutenant, United States Navy
B.S., Portland State University, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

ABSTRACT

The constant and fast-paced changes that are taking place in computer technology have brought forth a vast array of new applications. It is now possible to store not only standard alphanumerical data but also graphical, voice, and sound as well. This has opened up enormous possibilities for expanding the use of these data forms. This thesis is directed at exploring those possibilities and several current research projects that are attempting to model a multimedia database system. These models will be explored in terms of both their strong points and their weak points. Two possible applications will then be looked at in terms of how they could be modeled using each of these three models.

TABLE OF CONTENTS

I. INTRODUCTION.....	1
A. PURPOSE OF THIS RESEARCH.....	1
B. WHAT IS MULTIMEDIA?	1
1. Visual Information	2
2. Audio Information.....	3
3. Textual Information	3
C. CHARACTERISTICS OF MULTIMEDIA DATA.....	4
D. MULTIMEDIA APPLICATIONS AND IMPORTANCE	6
1. Instruction, Advertising	6
2. Supervision.....	6
3. Archiving, Information Retrieval	7
4. Publishing, Authoring.....	7
E. MODELING VERSUS IMPLEMENTATION	8
F. OUTLINE.....	9
II. DATABASE MODELS.....	10
A. WHAT IS A DATABASE MODEL?.....	10
B. TRADITIONAL RECORD BASED MODELS.....	11
C. CONCEPTS OF OBJECT-ORIENTED MODELS	15
D. CURRENT MULTIMEDIA DATABASE MODEL RESEARCH ...	19

III. ORION MULTIMEDIA DATABASE MODEL.....	21
A. ORION OVERVIEW.....	21
B. REQUIREMENTS FOR THIS MULTIMEDIA DATABASE	22
C. AN OBJECT-ORIENTED MULTIMEDIA DATABASE MODEL.....	23
D. SOME ADDITIONAL MODELING ASPECTS.....	27
1. Versions.....	27
2. Sharing.....	28
3. Access.....	28
E. ANALYSIS	31
IV. THE MINOS MULTIMEDIA MODEL.....	33
A. MINOS OVERVIEW	33
B. MINOS APPLICATION MODEL.....	35
C. MINOS USER INTERFACE.....	38
1. Browsing.....	38
2. Zooming.....	38
3. Narration	40
4. Transparencies	40
5. Process Simulation.....	40
6. Voice Segments.....	40
7. Annotations.....	41
8. Logical Text.....	41
9. Versions.....	41

D. ANALYSIS	42
V. EXTENDED RELATIONAL MULTIMEDIA MODEL	44
A. OVERVIEW	44
B. MULTIMEDIA REPRESENTATION FOR EXTENDED MODEL	45
1. Registration Data	45
2. Description Data	45
3. Raw Data	45
C. MULTIMEDIA DATABASE MANAGEMENT	46
D. THREE RELATIONAL SCHEMA TYPES FOR STORING IMAGES	47
E. FUTURE RESEARCH	48
F. ANALYSIS	49
VI. THE THREE MODELS COMPARED	51
A. INTRODUCTION	51
B. APPLICATION ONE	51
1. ORION Model	52
2. MINOS Model	55
3. Extended Relational Model	57
C. APPLICATION TWO	60
1. ORION Model	62
2. MINOS Model	64
3. Extended Relational Model	64

VII. SUMMARY AND CONCLUSIONS..... 66

A. AREAS OF FURTHER RESEARCH..... 66

B. CONCLUSIONS..... 66

LIST OF REFERENCES..... 68

BIBLIOGRAPHY..... 70

INITIAL DISTRIBUTION LIST..... 71

LIST OF FIGURES

Figure 1	Multimedia Elements	2
Figure 2	Definition of Nodes and Arcs.....	24
Figure 3	Document Sharing.....	29
Figure 4	Logical Object Aggregation Hierarchy of MINOS.....	36
Figure 5	Page Display With Menu in MINOS.....	39
Figure 6	Abstract Data Type Image	47
Figure 7	Relational Schema Types for Storing Images.....	49
Figure 8	Partial Photo Hierarchy	53
Figure 9	Message-Passing Protocol.....	54
Figure 10	Representation of Reconnaissance Photo and Ships.....	63

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my thesis advisor, Professor C. Thomas Wu, for his patience and assistance in guiding me toward the completion of this work. I would also like to offer a special thanks to my second reader, Klaus Meyer-Wegener, without whose personal sense of excellence and dedication this thesis would not have reached its final level of excellence.

I. INTRODUCTION

A. PURPOSE OF THIS RESEARCH

The constant and fast-paced changes that are taking place in computer technology have brought forth a vast array of new applications. It is now possible to store not only standard alphanumeric data but also graphical, voice, and sound as well. This has opened up enormous possibilities for expanding the use of these data forms. This thesis is directed at exploring those possibilities and some current research projects that are attempting to model a multimedia database system. This thesis will concentrate primarily on the database model itself vice the storage devices, input/output devices, or database implementation.

B. WHAT IS MULTIMEDIA?

The concept of multimedia is not new. It has surrounded us in our daily lives for years, starting from our first years in school, where we were surrounded by crayons, paints, pencils, books, and films, to adulthood, where we are interacting daily with television, radio, and books [Ref. 1]. What is new is the concept of multimedia combined with computers. The ability to link together sound, text, and video via a computer opens up enormous possibilities in both the academic and business communities. The exploitation of multimedia with computers will expand our ability to analyze information and achieve the realization of new and innovative ideas.

The elements involved in multimedia can be viewed as shown by Figure 1.

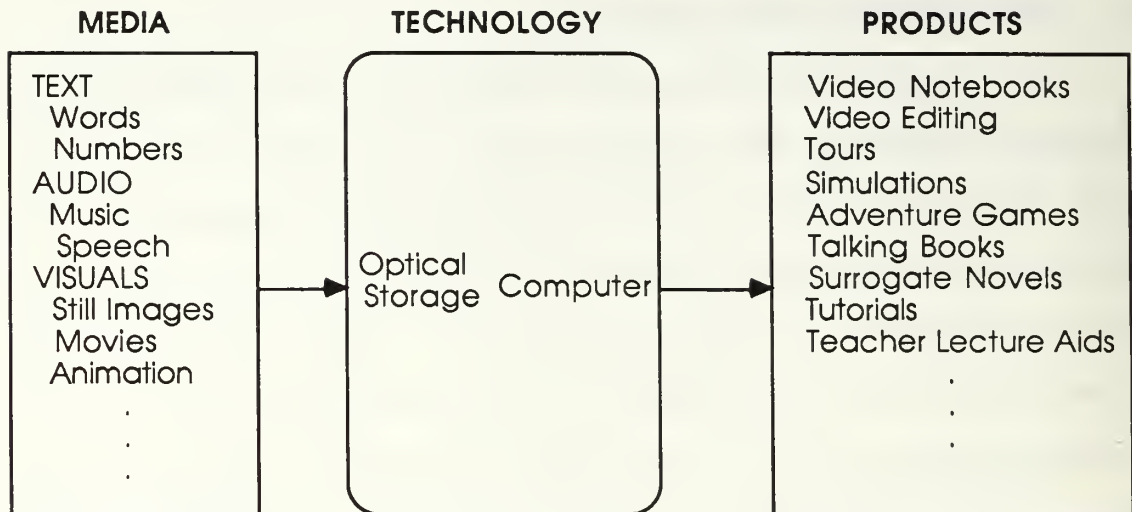


Figure 1. **Multimedia Elements**

The use of multimedia in a database environment would enrich the products in this list by allowing the development of products that could be used for military applications, management, and as decision aid tools.

1. Visual Information

There are several different choices regarding how images are stored. The decision must be based on how the image will be used. The formats available are video, bitmap, and vector. The video format is good primarily for use with moving images. The bitmap format stores an image as a two-dimensional grid of points. This format requires an enormous amount of storage space. Using the vector

format, the image is stored as a collection of shapes at specific positions, which requires less storage space than the bitmap.

2. Audio Information

Sound is recorded in digital format by measuring and digitizing the volume level at a very high rate. The interval of time between two measurements and the resolution will determine the quality of the recording. Each system must decide on the quality desired from the audio signal and the methods it wants to use to compress the large amounts of data involved.

3. Textual Information

The storage of textual information is well established and is usually handled by storing the character strings in American Standard Code for Information Interchange (ASCII). Another code often used to encode characters is called Extended Binary Coded Decimal Interchange Code (EBCDIC).

The above discussion looks at media only in terms of storage. It is also important to understand media in terms of the user's point of view. The user interfaces with his workstation in terms of what he can see, hear, speak, point to (by mouse, etc.), and type. Not all workstations provide the capability for the user to interface using all of these methods, but at least one will be available. The user can view text or, on systems with graphics capabilities, view images. On some more-advanced systems, there is even an ability to interface with the system using speech. The user is primarily unaware of the storage aspects of

his data; he is only aware of it in terms of these three interface possibilities.

C. CHARACTERISTICS OF MULTIMEDIA DATA

The media about which we have been talking comes in the form of textual, audio, and visual information. Each of these media types presents its own problems with respect to how it will be stored and how it will be efficiently retrieved. The storage of formatted data is not a new issue and has been successfully dealt with through past experience. The technologies involved in the storage of both sound and visual images are fairly new and therefore not refined to the same degree as formatted data. Nonetheless, it is the integration of all these information types into one homogeneous system that is the ultimate goal of a multimedia database system. The current level of technology does not allow us to handle the same kinds of queries that are already handled on traditional formatted data. There is, as yet, no proven way to provide the intelligence necessary in analyzing a picture in order to come to conclusions about what it presents.

What is it that we actually have to store? Is it just raw data in the form of strings of bits, or characters, or pixels? No, multimedia data always comes with additional data to interpret this raw data—this data is called registration data. Multimedia data also comes with operations like capture, edit, and search. The concept of searching, editing, or capturing multimedia data is not the same as for standard formatted data. For example, the complexity involved in searching for a particular image is no easy task and has prompted several different

approaches, as pointed out later in this thesis. Not only is the ability to search directly for an image without some sort of help inefficient but the computer cannot identify what a picture contains the way a human can and so needs further assistance. For example, take the image of a group of ships. How do we know what types of ships are there; how do we know the different aspects of the ships; how do we know whether they are friendly, and so on? Say we have an image where two submarines are moving at high speed, how do we know whether one is chasing the other one, and if so which one is the aggressor?

Of the three research projects examined in Chapters III, IV, and V of this thesis, there are three different ideas as to what a multimedia database is. In the research done by Christodoulakis, et al., multimedia is handled in terms of messages and documents made up of attributes, images, text, and voice types of information [Ref. 2]. In the work done by Woelk, et al., multimedia information is seen in terms of objects and is viewed as images (vector or raster, still or moving) and audio [Ref. 3]. Finally, in the work done by Meyer-Wegener, et al., multimedia information is organized in attributes of relations that are defined over domains like text, voice, graphics, and sound and signal data [Ref. 4]. The model discussed in Meyer-Wegener's paper uses raw, registration, and description data to simplify a contents-oriented search. Although there may be a great deal of similarity between these views, there is still no one precise, agreed-upon definition of multimedia information.

As pointed out by Woelk, et al., advances in technology have provided a means by which a great deal more information can be stored in a computer system. Now some real-world objects like books, newspapers, documents, and movies can actually be stored in the computer, rather than just having a name used as a reference for them. The idea is not just to store the real world object and let the user make changes to it but to actually understand the real-world object and utilize the computer system to execute that function more effectively [Ref. 5]. Bringing this idea to fruition will require advances in many areas of computer science, namely, artificial intelligence, DBMS design, and the design of the user interface.

D. MULTIMEDIA APPLICATIONS AND IMPORTANCE

The variety of applications that could become a reality using a multimedia database is limited only by the imagination. Four major application areas can easily be identified.

1. Instruction, Advertising

The enrichment of the learning experience through the use of multimedia tools will allow both teachers and students alike to more vividly communicate about their topics. Both of these areas have a static data set that is very active, which means that the system guides the user, draws his attention to something, and alerts him as necessary.

2. Supervision

The area of supervision can cover a wide variety of possibilities, for example, supervision of a battlefield or supervision of the

command information center on board a combatant ship. The data set is usually dynamic in these types of applications and is very active. The applications for a multimedia database in this area could include administrative uses, maintenance and repair, training, and tactical decision aids.

3. Archiving, Information Retrieval

The data set here is static and not very active, which means that this is a passive system which waits for commands and does not guide the user. The ability to archive large amounts of information is essential, especially in any multimedia application.

4. Publishing, Authoring

The data set here is very dynamic due to the need for constant editing, but it is not very active. This application area can easily overlap any of the others because the need to produce reports, et cetera, is necessary in the academic community and in the business and military environments.

The use of multimedia will allow the decision makers of tomorrow to analyze information in ways that were previously not available to them. Currently, the use of a multimedia database for real-time operations is limited by its lack of speed, but with future technological breakthroughs this will change. In the meantime, it is important to capitalize on what is currently available. Multimedia database research is important because through its use we can enrich our ability to communicate and our ability to make sound decisions.

E. MODELING VERSUS IMPLEMENTATION

The real strength of a database system is its ability to manage data. Database systems have been relatively successful when dealing with routine formatted data, but with the addition of unformatted data it is questionable just how much actual management of the data is really occurring. Implementations that provide simple storage and retrieval of this data may not be enough, depending on exactly how complicated the users' requirements are for the data management.

It is important to introduce the concept of a multimedia database management system (MDBMS) at this point. What is this system supposed to do? It is supposed to manage the data, which means it will primarily handle the storage and retrieval of the data. This is no simple task because multimedia data consists primarily of unformatted data. What the MDBMS does is defined exactly in the database model, i.e., the objects it manages (documents, relations, and tuples) and the operations on them. Currently there is no agreement on just exactly what a MDBMS should look like, and so it follows that there is also no agreement on what a MDBMS should do. The multimedia database management system cannot do all the applications outlined in Section D by itself. A user interface built on top of the MDBMS will need to provide the additional requirements for a given application.

It is an extremely difficult problem to design a database model that is capable of handling the variety of data types encompassed by the term *multimedia*. This research is still in its very early stages and has primarily been directed at small, restricted application areas. Even

if a model is designed that appears to address the variety of problems unique to dealing with multimedia data types, like the complexity involved in handling unformatted data and the analysis and reasoning that are involved, there is no guarantee that the model will be feasible to implement. Two possible approaches to this dilemma are to restrict the model and its implementation to a small specific application environment or to try and adapt or extend an existing model that has already proven itself to be very reliable. There are other approaches, as well, but they are much more difficult than the two suggested above.

F. OUTLINE

The following chapters first discuss the traditional record-based models and the object-oriented model and then examine, in some detail, three of the currently proposed models for a multimedia database management system. Each project will be looked at individually and the strong and weak points of each will be identified and analyzed. With this as a foundation, we will look at two possible applications and how they could be modeled using each of these three models.

II. DATABASE MODELS

A. WHAT IS A DATABASE MODEL?

A database model is defined as "...a convention of specifying the concepts of the real worlds in a form understandable by a DBMS." [Ref. 6] The role of a Database Management System (DBMS) is to map the stored view to the actual physical structure—in other words, to implement the data model [Ref. 6]. The DBMS must provide at least the following as a minimum:

1. persistence
2. concurrency
3. consistency
4. security and authorization

Database management systems have been developed mainly to manage formatted data rather than information because the information is usually embedded in the data. We must develop a more powerful data model that will allow us to capture the more complex data representations that make up multimedia information. [Ref. 7]

A multimedia database model must represent the variety of concepts that are included in the definition of multimedia information in a way that can be understood by a DBMS. This becomes a very complex problem to deal with using the technology currently available. The key issue here is that current models are not strong enough to describe everything we need to describe in multimedia data. In this chapter,

traditional database models will be looked at to see how they apply to the design of a multimedia database model and what problems will need to be addressed and overcome.

B. TRADITIONAL RECORD BASED MODELS

Traditional database models using records have, in most applications, been efficient, well understood, and generally easy to use. There are several clear advantages to record-based models that should be pointed out. Ideally, we want to maintain these advantages and yet, at the same time, go beyond them.

1. They are easy to understand.
2. They are easy to implement efficiently.
3. They have a well-established theoretical foundation (this is true specifically for the relational model).
4. They are currently the industry standard.

A *record* is taken to mean a fixed sequence of field values that conform to a static description contained in programs or catalogs. Looking at using a record-based model in modeling multimedia information, it rapidly becomes apparent that this model's ability to represent the types of information that make up multimedia information is incomplete. As Kent pointed out in his paper, records are an excellent tool for processing information as long as that information fits into a certain pattern [Ref. 8]. Multimedia information in its entirety does not fit this pattern and the information that does fit into the record structure has to depend on special-purpose application programs to supply the supplementary information necessary to process the data.

There is a great deal of ambiguity in trying to come up with a rule as to what can and cannot be represented in a record structure and just how certain pieces of information will be represented when there is more than one choice available. For example, say you wanted to define a record type that would contain the location and owner of all buildings in a given district. The record could be called the "Owned By" record. This seems simple enough, but what if building owners in that district are a combination of private citizens and companies. It may not always be easy to determine whether the owner you have identified is a private citizen or a company. At this point, you could add another field that identifies whether the owner is a private citizen or a company and then have one field for the private citizen name and one for the company name. There are partial solutions, as you can see, but they usually involve a cost factor of some sort and move the data structure even further away from the semantic structure of the relationship that is being modeled [Ref. 8]. Taking this point even a little further, let's look at how this would apply to multimedia. For example, let's take a "Shows" relation for images. If the image contains several different things, like a ship, an aircraft, and ship's personnel, then there could be a problem. The Shows relation combines the object-id and the image-id (or key).

[image 1, ship 1]

[image 1, aircraft 1]

[image 2, ship 2]

Just as in the case of Owned By, we could use Shows (image-id, select, ship-id, aircraft-id,) with tuples like these.

<image 1, ship, ship 1, NULL,>

<image 1, aircraft, NULL, aircraft 1,>

It can be seen here that the cost factor mentioned earlier is in the use of the NULL value. This same problem would come up in trying to use a "Talks Of" relation for speech.

The use of a record structure assumes that there will be both vertical and horizontal homogeneity in the data. This means that a given field will contain the same kind of information for each record and that each record of the same type will contain the same fields. Records are designed to be used when the entity that is being represented has the same kind of attributes so that all instances will require the same fields [Ref. 8]. The question for the representation of multimedia information is whether this requirement really applies. It is very difficult to deal with the rich semantics contained in multimedia data, so the restrictive semantics associated with traditional DBMS and the formatted data it dealt with are no longer sufficient. As an example, look at the difficulties involved in dealing with only one contributor to multimedia information, i.e., images. The internal format is different from image to image. The inhomogeneity of multimedia data also shows itself in the variable length and the strong variability on structure of the data.

Even just trying to represent something like a ship textually can be a problem because although an aircraft carrier and a submarine are

both ships, there is a considerable difference in the facts that are relevant to each one. While it is obvious that an aircraft carrier has a flight deck, it is also obvious that a submarine does not. It would therefore be very difficult to define a conventional ship record type due to its not having homogeneous attributes over its set.

Kent does suggest some possible solutions for dealing with information that is not homogeneous but, as he points out, these solutions can be cumbersome and inefficient. These solutions involve either allowing null values in many fields or letting the same field have different meanings in different records. [Ref. 8]

Whether you are trying to solve the problems associated with vertical or horizontal homogeneity, there are drawbacks in all possible solutions. The bottom line is that the record structure is not going to be well suited for representing information that has problems with vertical and horizontal homogeneity. Therefore, the traditional record-based model is not able to adequately and completely represent multimedia data.

Representing a relationship in a record-based model is not as simple as it might first appear. There are many ways that this relationship can be identified. The most common modeling technique for a many-to-many relationship is to take the identifier from each of the two entities and pair them together in one record. Not all relationships are represented directly because some relationships are implied by other relationships. For example, if certain mission types are delegated to certain ship types and if each weapon system on a

ship is directly involved in its mission, then to find out whether a certain weapon system is involved in a certain mission type you must match the ship's ID number in the Ship record and the Mission record. In other words, a weapon system belonging to a certain ship type and a mission being assigned to that ship type indirectly imply that the weapon system is connected to a given mission.

These are only some of the areas where record structures are unable to model the semantics of the information in a way that is clear and complete. There are ways to compensate for some of these limitations, but there is always a cost associated with each remedy.

In conclusion, the traditional record-based model was designed to handle formatted data and, since multimedia data also refers to unformatted data, the pitfalls associated with the record-based model become magnified. Extending the traditional record-based model can allow it to be used to represent multimedia data but, even so, many of the same problems outlined above will still remain. The decision as to whether these problems are acceptable will depend on the application requirements.

C. CONCEPTS OF OBJECT-ORIENTED MODELS

The term *object-oriented* can be very confusing because its definition is dependent on the user's concept of what it should mean. In an article by O. M. Nierstrasz, the author tries to capture some commonality from all the object-oriented definitions in use and then tries to focus on this aspect in discussing object-oriented concepts. The common quality he sees in the variety of uses of the term

object-oriented is the idea of encapsulation [Ref. 9]. This means that the object gives the appearance of being encased or in a capsule and can only be accessed through a set of functions, operations, or, as they are now called, messages. Encapsulation is predominant in object-oriented systems in that programming is done in terms of objects, which are what hold the values, as opposed to data, which are the actual values themselves.

The major characteristics associated with object-oriented systems are class inheritance, polymorphism, and instantiation via object classes. It must also support both aggregation and generalization hierarchies. Nierstrasz feels that to require a programming language to meet all of these requirements in order to be called object-oriented is too restrictive. Instead, he suggests that any programming language that exploits encapsulation to any degree is in fact object-oriented. This allows the discussion to be directed at how object-oriented a language is—in other words, the degree to which it is object-oriented, not simply to whether it is object-oriented.

Some of the fundamental building blocks of an object-oriented language are class, object, and message. The class defines the structure and behavior of its instances. A class is something like a relation. The class defines the attributes (instance variables). For example, an instance of the Officer class might be LT Smith. By being an instance of the Officer class, it already has a set of methods and attributes associated to it. Each member of a given class will behave the same because all will respond in the same way to the same message. The

encapsulation can be seen here because objects that are similar can be grouped together and can be surrounded by the structural and behavioral aspects of all their members.

Programs are made up of objects. An object is something like a tuple and is what actually holds the values. An object can represent a tangible or intangible real-world object. An example of a tangible real-world object in a military administration application might be "Officer," whereas an example of an intangible object might be "Schedule."

A method corresponds to the concept of a procedure or a program. It is a procedure executed inside an object to change the state of the object (update the data) or to report the state (read the data). An object receives a message and then selects a method and executes it. For example, the message might be to give the rank of an officer. Provided there is an instance variable called Rank, the value for it will be returned. If the classes of Officer and Enlisted are grouped under a superclass called Soldier, then many of the procedures used by both subclasses need only be written once because these procedures are then inherited by these two subclasses. Both methods and messages exist in the relational model; for example, the update of an attribute is a method applied to a tuple but they are fixed, not user written.

The generalization and aggregation abstraction concepts are a necessary part of the object-oriented model. Generalization allows for the creation of a hierarchy; for example, "Person" is a generalization of officer, employee, and pilot. "Officer," "employee," and "pilot" are

all types of persons. Aggregation is when an object is an aggregate of other objects; for example, an "Automobile" is an aggregate of engine, tires, and body frame. The engine, tires, and body frame all go into making up the object Automobile.

Inheritance allows for the reusability of code by allowing objects in a subclass to share behaviors that belong to their superclass. This is one of the major advantages of the object-oriented model because it allows for a reduction in redundancy. This is very important when dealing with the large amounts of data associated with multimedia, especially as objects take on additional properties and behaviors.

Polymorphism is used to describe the situation where the same message is sent to different objects but can illicit different responses depending on just who the receiver is. Without this ability, the design would need to incorporate a structure to handle every possible situation. For example, if you wanted to send the message Compute Pay to both Officer objects and Civilian Employee objects, you would need to have a Compute Pay One and a Compute Pay Two because the way that pay is computed for these two objects is different. Polymorphism allows the same message to be sent to both objects but to be handled differently by the receiver. Obviously, this helps to reduce the amount of code and so is very important.

In the previous section, an example was given that pointed out some of the problems that the traditional record-based model would have in trying to represent something like a ship. In comparison, the object-oriented model would have little trouble doing this because the

object Ship could be used to contain all the common qualities for different types of ships and then the specific types like aircraft carrier and submarine would be objects in a subclass that would inherit common qualities from the superclass Ship.

In summary, the object-oriented model allows objects in the real world to be represented in a way that is more closely tied to the way the user thinks of them. The classification and methods allow the many different aspects of multimedia to be easily captured. One of the advantages to the object-oriented approach is that because of polymorphism and inheritance you can usually limit the scope of the changes to modules. The basic qualities outlined above that apply to the object-oriented data model are what make it so well suited for meeting the requirements of a multimedia database. One of the biggest disadvantages of an object-oriented DBMS is that, due to the increased implementation complexity that comes with the increase in user friendliness, the system is usually much slower. The biggest advantage of an object-oriented DBMS is that it allows for a representation method that aligns itself more closely to a natural way of human thinking.

D. CURRENT MULTIMEDIA DATABASE MODEL RESEARCH

Traditional DBMS technology has been directed toward commercial applications that deal primarily with formatted data, but information in the true sense comes in both formatted and unformatted forms which include image and audio. The advances in DBMS technology must direct themselves toward dealing with the increases in

complexity introduced by the addition of unformatted data in an efficient manner. There are database model projects like POSTGRES, IRIS, and EXODUS that are currently in progress. Although they are not specifically aimed at multimedia, they do focus on extensibility, and by doing so they do try to cover the multimedia aspect.

III. ORION MULTIMEDIA DATABASE MODEL

A. ORION OVERVIEW

The first paper to be examined is one dealing with an object-oriented approach to the design of a multimedia database. The data model described in this research paper was implemented using a prototype called ORION. It was implemented in Lisp and run on a Symbolics 3600 Lisp Machine [Ref. 5]. The example used in this paper is a simple memo-type document. The information normally contained in a memo can be broken down so that it forms an aggregation hierarchy. It is the abstraction provided by the aggregation hierarchy that allows the integration of the different types of multimedia data. It is the responsibility of the database system to support this hierarchy. It is the flexibility and generic nature of the aggregation hierarchy that makes it so suitable for representing multimedia type information. [Ref. 3]

The generalization hierarchy also becomes important when the memo is broken down into its smaller parts. The generalization hierarchy allows one part to be a subtype of some other part so that the subtype can inherit the properties of that part. With the extremely large number of objects that could be a part of the multimedia database, the property of inheritance is critical.

B. REQUIREMENTS FOR THIS MULTIMEDIA DATABASE

This paper outlines the 15 areas that its authors feel are necessary to provide a multimedia database application [Ref. 3]. Many of the areas identified are necessary in providing a database management system and are not specific to a multimedia database system. These requirements are also specific to this model; a more general model would require some changes.

1. The database must support the aggregation hierarchy.
2. The database must support a generalization hierarchy.
3. The application must be able to maintain data in separate relations which will represent the data that is a part of every memo.
4. The properties of an object can be represented as either data or as a procedure.
5. A schema for the body of the memo would be difficult to define because the body can take on many different looks depending on the user's viewpoint, so the user will provide feedback which will be used to determine the necessary constraints for the body. These rules will be represented using procedures.
6. The database must support the ability for any node to have a relationship with any other node since each memo is different and will require this sort of flexibility.
7. The semantic information that is needed to dynamically modify the memo schema must be maintained by the application.
8. The database must be able to manage the interactive modification of the physical presentation of the memo information.
9. The database must provide support for the creation, control, and change notification of versions of documents.
10. The database must provide some mechanism that will control the concurrent access of the same data by multiple users.

11. The database will need special functions that will minimize the movement of the large amounts of data involved in unformatted data-like images.
12. The database must support the snapshot and view mechanism so that graphics can be used to represent the dynamic changes in the underlying data.
13. The database must provide the ability to share data among several different documents in order to reduce the amount of storage needed.
14. The database will need to provide access to documents based on some association that is common among them.
15. The database must provide some mechanism for recovery of the secondary storage and some type of transaction recovery.

C. AN OBJECT-ORIENTED MULTIMEDIA DATABASE MODEL

The approach taken here is to represent the complex and varied types of data as token objects. These token objects provide generalization and instantiation and also aggregation. This means that the token object will represent all the token objects that are below it in the aggregation hierarchy. There is one substantial difference in the design of this token object over the traditional object design—the token object can be stored independently and so can be used by more than one aggregation hierarchy. [Ref. 3]

There are six different kinds of nodes used in this model, as represented in Figure 2. [Ref. 3]

1. Token objects representing classes
2. Token objects representing instances
3. Relationship objects
4. Attributes

5. Method objects

6. Intrinsic data objects

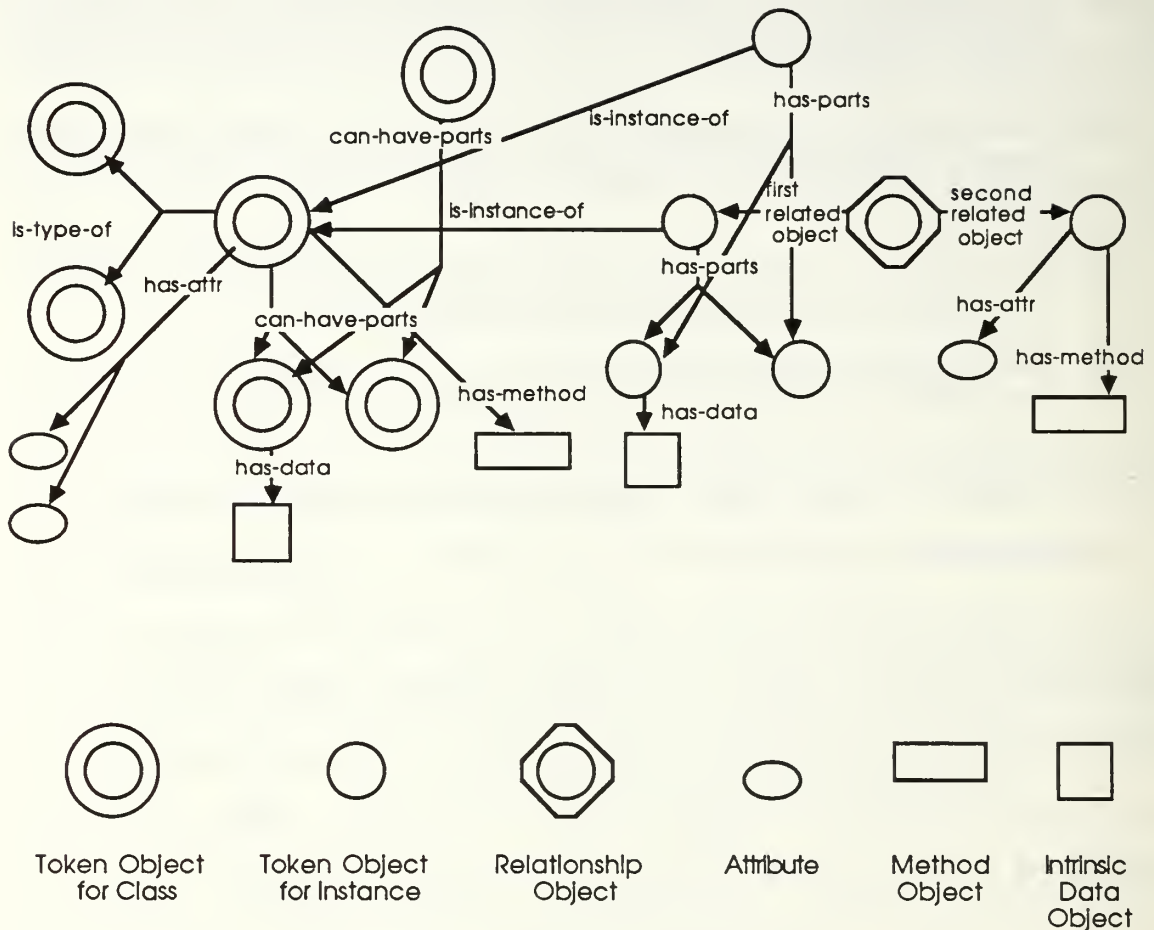


Figure 2. **Definition of Nodes and Arcs**

The token object is intended to provide one mechanism that is able to represent the many different types of data necessary for multimedia applications and the relationships between these data. For example, token objects are used to represent an instance of an object. This is done by connecting a directed arc between the token class and the token instance. [Ref. 3]

Generalization finds itself represented in this model by using an Is-Type-Of arc between an object class and a token object. For example, an Is-Type-Of arc would exist between the Department Memo class and the Memo class, showing that the department memo is a type of memo. This allows the department memo to inherit the properties of the Memo class and have its own properties as well. This type of containment relationship between a higher-level object and the objects below it is what allows for inheritance to occur. Inheritance is a way of sharing methods; by doing so a great deal of redundancy is spared and so efficiency is improved. [Ref. 3]

The ability to implement aggregation is critical in meeting the needs of a multimedia database system because aggregation is an abstraction method that allows relationships to be treated as higher-level objects. This allows for the expression of relationships among relationships [Ref. 10]. It should be pointed out here that the relationship that establishes aggregation is not the relationship object shown in Figure 2. The aggregation hierarchy that is created is used for data sharing, database access, and version control. An aggregate token object is made up of the simple token object and all of the token objects below it in the hierarchy. This is useful because object properties are then able to be shared down the hierarchy. For example, in a documents application a Body instance could consist of several Paragraph instances; these Paragraph instances would then share the properties of the Body. [Ref. 3]

A token object can have attributes. Each token class can be described by its attributes and its methods. An example in this model is the font size 12, which is identified as an attribute of the memo class. This is represented in this model by using a directed arc called Can-Have-Parts. [Ref. 3]

Each token object has operations or instructions that can be performed on it which are called *methods*. These methods belong to the class and so are shared by each of the class's instances. A method is very much like a procedure or a subroutine, except that it is specific to a certain class of objects. The only way to gain access to an object's data is through use of its methods. For example, a light switch is a method for lighting. In order to gain access to lighting a room you must turn on the light switch—in other words, you invoke the lighting method.

In order to provide a mechanism to handle the other relationships among objects that are not handled by the generalization, instantiation, and aggregation relationships, this model uses what it calls a Relationship object. The user can name these relationships as needed. These Relationship objects are intended to provide functions for things like annotation of text by sound. [Ref. 3]

The intrinsic data objects are intended to hold the multimedia data. In most cases, they are really no different than special attributes. [Ref. 3]

D. SOME ADDITIONAL MODELING ASPECTS

In order to support this model, certain functions will be necessary. First of all, the creation of token objects is needed in order to create an object class or an object instance. The Is-Type-Of arc allows for classes to be created as specializations of other classes, and classes can be linked to one another using Can-Have-Parts links. The use of inheritance will reduce the work required to create new classes. [Ref. 3]

1. Versions

Version control is important in a document environment. The creation and control of versions are handled using versions of the token objects. This paper discusses both historical versions and alternative versions. Historical versions will represent the history of the document as it is changed over time, whereas alternate versions will show different implementations of the same object. [Ref. 3]

If modifications are required for a given paragraph in the text of a paper, a copy of that paragraph instance is made and then the changes are made to the copy. Once this is done, the author can specifically request that a new historical version of the paper be created. Alternate versions do not represent sequential changes to the document as is the case for the historical version; they represent completely different representations of the same document. Each alternate version of the paper can have its own historical versions. [Ref. 3]

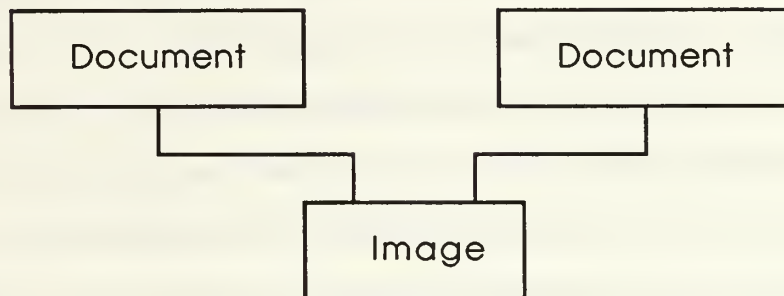
2. Sharing

There are two ways to handle the sharing of token objects, as shown in Figure 3. If the changes that are made to the token object are to be seen at a later time, the user will need to request a reference to that image. By doing this, a reference token object is created which then links the objects into the aggregation hierarchy of the memo. If the user doesn't care about the changes made at a later time, then a deferred copy is used. If changes are made to the shared image, then a copy of the image is made for whoever made the changes. The original is left unchanged for others who share it. Making an actual physical copy, which often requires replication of large amounts of data, is thus postponed to the last possible moment. This provides the user with the appearance that a copy of the object has been made. The deferred copy does not affect the view of two different images—it simply avoids copying as long as possible. [Ref. 3]

3. Access

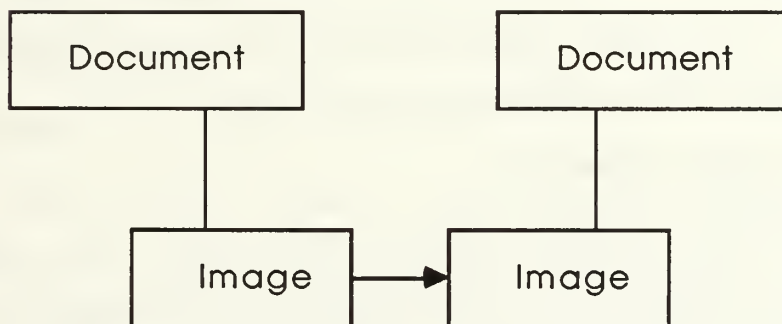
The intent in this model is to use the associative access available in instantiation, generalization, and aggregation. Access can be made to an object instance via its class. By taking the inverse of an Is-Instance-Of arc that is directed from the instance to the class the instances of that class can be accessed. If an aggregate token object is being dealt with, the descendants can be accessed using the Has-Parts arc. The user can create relationships among objects using relationship objects and these can be used to move from one object to another. The relationship object serves as a link between two object

instances. Object access can also be gained using the generalization hierarchy. [Ref. 3]



Reference

The image must be up to date in both documents.



Copy

The image must be stable in both documents.

Figure 3. **Document Sharing**

By using one or more of the different access modes available, a query can be created. An example query given in Woelk's paper is "Retrieve memos which contain pictures of database machines and which reference the 1985 Database Program Plan." [Ref. 3] This query is made up of subqueries because all memos containing pictures of database machines must first be located and then this group of memos must be searched for those that reference the 1985 Database Program Plan. It is important to note that the order in which these subqueries are executed will cause the performance to differ. Several different access modes must be used. The first access would be made to the memo instance via the memo class by taking the inverse of the Is-Instance-Of arc. This will provide the identity of all the instances of the Memo class. For each Memo instance, a request for descendants in its aggregation hierarchy can be made using the Has-Parts arcs. In this example, we need to use the Has-Parts arcs to gain access to the Body instance and then again to gain access to the Image instance. At this point, it must be determined whether the image contains any pictures of database machines. [Ref. 3]

Multimedia applications require that the multimedia information can be shared and manipulated. Woelk and Kim address this requirement in their paper describing the implementation of the Multimedia Information Manager (MIM) in ORION, which is responsible for the management of the I/O devices in the DBMS. The paper outlines the three major design objectives used to support their goal

and how these design objectives were implemented. The three design objectives are extensibility, flexibility, and efficiency. [Ref. 11]

Extensibility allows the system to be extended to support new multimedia devices and new needs that are identified at some future time. MIM provides for this extensibility by using object-oriented concepts. The flexibility of the MIM comes in its ability to store, present, and control multimedia objects with an internal format that is spatially oriented like a bitmap and multimedia objects with a sequential internal format like text. Efficiency is achieved by allowing for the storage blocks to be shared among multiple versions of a multimedia object. Another contribution to the efficiency of the system is made by optimizing the transfer of data. Since the amount of data involved in multimedia applications is usually large, this is very important. The MIM eliminates unnecessary buffering of data by transferring information from a storage device directly to the presentation device whenever possible. [Ref. 11]

E. ANALYSIS

This section takes a look at how well this model meets the requirements of a multimedia database system. One of the most important aspects of a realistic data model for a multimedia database is how easily and efficiently it can be implemented as well as how friendly querying the system will be to the user. This paper has not really explored these issues. The implementation of the concept of a relationship object as outlined can be very complex, depending on the power of the query language and whether you want to do more than

just navigate around. The object-oriented approach does seem to be a natural way to represent a multimedia database, but it is important that this ease of representation be accompanied by a query language that is one the user can use in a way that is both familiar and responsive. The issues of speed and real-time processing have also not been addressed here. Although these features may not be critical to this particular application, they should still be considered in the overall design of a good multimedia database model.

IV. THE MINOS MULTIMEDIA MODEL

A. MINOS OVERVIEW

The second paper to be examined deals with an object-oriented multimedia information system called MINOS. The implementation of the MINOS model is accomplished using a Sun 3 workstation running Unix. This system is designed to view attributes, voice, image, and text as multimedia input. A magnetic disk is used to store the text, attributes, and images and an analog device is used for voice storage. MINOS provides the ability to view multimedia input, browse through it, and extract selected information. It also allows for the sharing of information between multimedia documents. The system is designed primarily with office applications in mind. In essence, the system is designed to store all information in large-capacity devices and then to extract information from it in order to use it to create new documents. The system is seen as a network where managers can communicate quickly and interactively with one another using multimedia information. [Ref. 2]

A possible scenario for the use of the MINOS system as described in this paper is as follows. An office worker can originate a query and then browse through the documents that qualify under that query, looking for the desired document. When a document is located that has some relevance to the task, the page-browsing interface can be used to take a closer look at the document. If information that can be

used is found, then the extraction interface can be used to pull the information out. After all the information desired has been extracted, a comparative interface can be used to compare pieces of the extracted information in order to select the most relevant information. Then the extracted information pieces can be put together to form a new document. The user can also create new information and merge it with the information that has been extracted to form a new document. [Ref. 2]

Multimedia documents can be in either an archived state or an editing state. The difference is that archived documents cannot be edited—they can only be deleted. The MINOS information system allows for presentation, browsing, information extraction, information sharing, and formatting. Queries are made on the text part, image part, voice part, and attribute values. The voice part is queried primarily on whether a voice part exists for that document. The image part is queried on some textual portion of the image, or some similarity relationships among the image objects. The user can interface with the archiver using three methods:

1. Query specification interface
2. Browsing through documents interface
3. Browsing within documents interface

The first item helps the user to make a query through the use of a menu and some graphics. The browsing interface provides small iconic representations of a document in order to help trigger the user's memory when searching for a specific document. The browsing within interface allows the user to look inside the document at

specific information. The user can use any part or all of the document to create another document. [Ref. 2]

The user has enough flexibility to extract information and then combine it with the information he creates to form a brand new document. The system provides higher-level commands so that non-programmers can create even more complicated documents without worrying about the implementation details. [Ref. 2]

B. MINOS APPLICATION MODEL

MINOS is based on an object-oriented model. Each object has its own unique identifier and can have its own set of attributes, methods, and data. This model includes the abstraction concepts of aggregation, generalization, inheritance, and versions. [Ref. 2]

MINOS uses two models to describe the multimedia documents. The first is called the logical model and the second is called the physical model. The logical model basically represents the logical instance, which is a collection of logical object instances, and the physical model, which designates the presentation of the object, i.e., the document, on a given output device. The physical model is a description of the components of what will appear on the output device. There is a mapping from the logical to the physical to show which logical models are mapped to which physical models. Figure 4 shows the MINOS model of what a document object will look like. The double-sided arrows from the rectangles for attribute, image, voice, text, and annotation mean that a document can have zero or more of

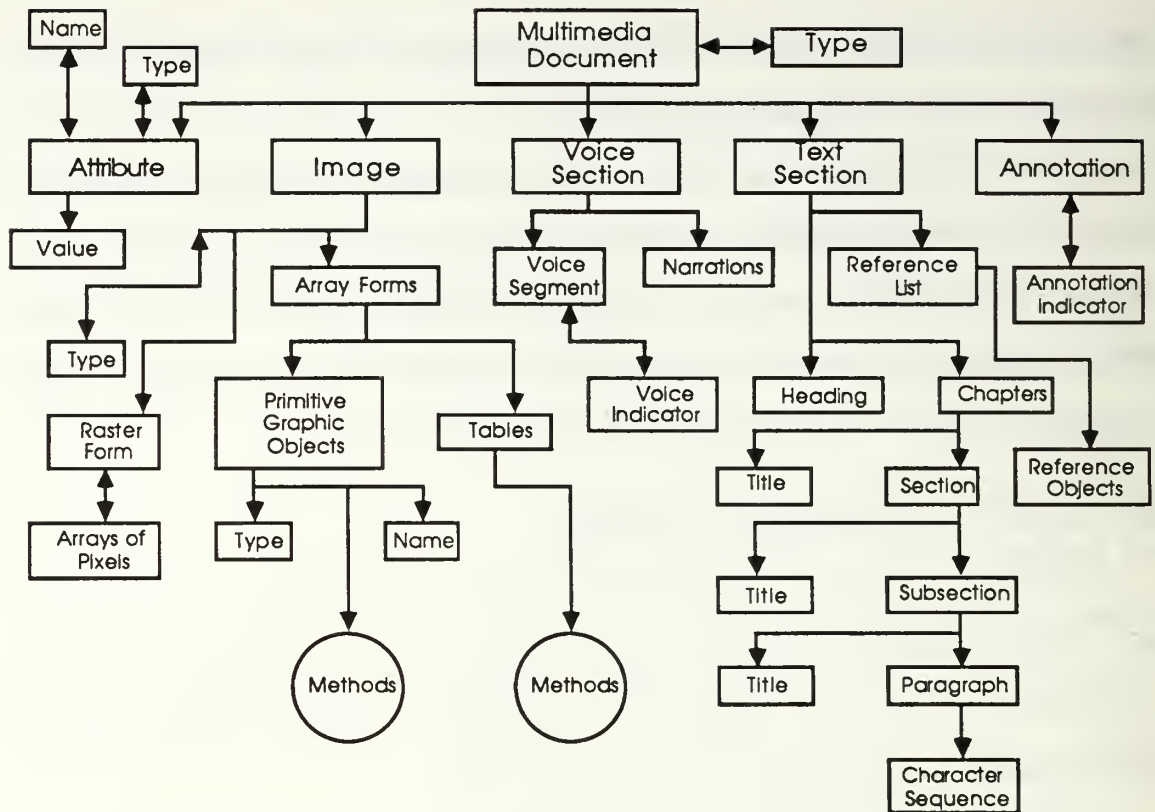


Figure 4. **Logical Object Aggregation Hierarchy of MINOS**

each. The attributes typically represent redundant information because it is information that has been extracted and then stored as an attribute value. If an attribute value needs to be changed, the document must be extracted from the archiver, and then the document containing the changes will replace the old document in the system. In choosing attributes, several factors need to be considered. An attribute that will have a low probability of having a null value, one that will be used frequently in user queries, and one that will only have very few qualifying documents make up the best candidates. In looking at

documents that are quite long, the cost involved in searching for them should also be a consideration. [Ref. 2]

The voice part of the multimedia document can have segments and narrations. A voice narration is associated with an actual page and will automatically be played when that page is entered, whereas a voice segment is associated with a specific paragraph or image and will only play if the voice indicator is selected by the user. [Ref. 2]

The annotations function as links between related multimedia documents. If an annotation indicator is displayed near a component, then, when selected, the first page of the related document will be displayed. [Ref. 2]

Images in MINOS will be made up of many objects and form several levels. The image type is used to identify sets of objects that contain a particular type of image. The images themselves can be composed using a raster form or an array form. The array form can be composed of tables and primitive objects which are graphic objects that have methods to support them. [Ref. 2]

A variety of relationships can exist within the document. A text object can have several images related to it or voice segments can relate to a given image object. The relationship instances are objects contained in the logical document instance but are not shown in Figure 4. [Ref. 2]

The MINOS model is implemented using two tables for each document—one for information about the logical structure and one for information about the physical structure. The logical structure would

encode the aggregation hierarchy for a document object and its object relationships. The physical table would contain the same type of information for the physical document instance. The MINOS system uses what it calls a "document descriptor" to encode the information in the physical and logical model in a compacted form. [Ref. 2]

C. MINOS USER INTERFACE

MINOS focuses on information that will primarily be used within the computer system and not necessarily require hard copy use. The capabilities presented to the user are browsing, zooming, narration, transparencies, voice segments, annotations, logical text, process simulation, and versions. These capabilities are built on top of the actual multimedia model. Figure 5 shows what an actual screen output would look like.

1. Browsing

The menu displayed on the right-hand side of the workstation will show the browsing methods applicable to the document being used. The menu options that are shaded are the ones that can't be accessed in connection with the current document. The page objects being viewed will be displayed on the left side of the workstation. [Ref. 2]

2. Zooming

The user is able to zoom in on an object by selecting the zooming option from the menu. Zooming can only expand on information that already exists in the image. [Ref. 2]


<p>TORONTO GENERAL HOSPITAL DEPARTMENT OF PATHOLOGY</p>  <p>PATIENT'S NAME: A. SMITH INSURANCE #: 123456 EXAM DATE: 4/6/85 DATE OF BIRTH: 29/2/50</p>	Return to Browse		Create Filter	Append to Filter
	Previous Page		Next Page	Select Page
	Mark Current Page		Zoom	Display Marked Page
	Select Annotation		Exit Annotation	
	Narration		Voice	
	Previous Chapter		Next Chapter	
	Previous Section		Next Section	
	Previous Image		Next Image	
	Grep First Word		Grep Next Word	
	Extraction			

Figure 5. Page Display With Menu in MINOS

3. Narration

A narration is a voice session that is associated with a specific page. When a user requests to see a page that contains a narration, it will automatically play unless the user selects the option to turn it off. [Ref. 2]

4. Transparencies

MINOS creates transparencies by declaring a page to be a superimposition of another page. So when a transparent page is brought to the screen, the page that preceded it is not deleted—the transparent page is just superimposed on top of it. Using the transparency and narration together can prove very useful in a presentation format. [Ref. 2]

5. Process Simulation

The appearance of a continuous processing of information can be obtained by specifying that once the first page of a set is looked at, the following pages will advance automatically. This allows the system to simulate animation without having to use a graphics programming language by using a combination of narration, voice, and process simulation techniques. [Ref. 2]

6. Voice Segments

If an object such as an image or a paragraph of text has voice information associated with it, a voice indicator will be displayed on the screen and the user can select it and have the voice segment played. [Ref. 2]

7. Annotations

Annotations are explanatory notes or additional comments of greater detail on a subject. Annotations are represented in MINOS as links to multimedia documents. If annotation is selected while viewing some independent document, the annotation document will replace that document as the current document. An annotation can also be an image. These annotation documents can be nested. The user can browse through them and, when ready, select to return to the original document. [Ref. 2]

8. Logical Text

This is textual information that will fit on the same page as the image or images with which it is associated. If the text is more than what will fit on one page, the selection of next page from the menu will just replace the textual part of the picture until it has all been displayed. The purpose here is to allow for continuous display of the image and its associated text so that the user is not required to continually go back and forth between pages. The implementation is accomplished by using the relationship between an image object and a text object, as shown in Figure 4. [Ref. 2]

9. Versions

Versions are necessary in MINOS just as they were in the ORION project discussed in the previous chapter. In the MINOS model there is only one parent version, but there can be any number of children versions, so what is created is called a version tree. An example of versions given in the MINOS paper is the use of a

document that needs to be presented in several different languages in order to meet the various client needs. If any changes become necessary, it is the version support provided by the system that would allow for all the document versions to be located in order for the change to be made. [Ref. 2]

D. ANALYSIS

This model does contain the necessary elements for a multimedia database in that it deals with images, text, and sound in the same environment. The system also provides an extremely user-friendly environment through its use of a menu-driven interface. But the design itself is limited in its scope because it deals with only one application area and the user is limited in his query ability by what is already available using the menu. The primary use of the functions provided seems to deal with the manipulation of data rather than the processing of data.

In the sense that the system is not generic, MINOS does not offer the power of a standard DBMS. The ability to represent reality using an abstract model is limited to things that can be modeled as documents. For example, it is not possible to define the object Ship using this system. There are many examples of scenarios where this model does not capture the semantics nicely. One might be a military application where there would be a need to capture a variety of incoming signals in different forms that are not able to be represented by documents. These inputs need to be integrated in order to provide assistance in

decision making by those in command. This model is not able to manage data in this manner.

Within the scope of its design, it does provide a tool that can be useful in an office or academic environment. The model does not lend itself to use in a real-time environment, largely due to the heavy CPU demands made by much of the multimedia data and the limitations of the hardware systems currently available.

The MINOS system does seem to support the concept that within current technology it is only possible to implement multimedia systems that are narrow and specific in their application.

V. EXTENDED RELATIONAL MULTIMEDIA MODEL

A. OVERVIEW

The third research paper to be examined explores the use of an extended relational model as the means for modeling multimedia. To do this, it extends the relational database by adding new attribute types to it. The example used in this paper uses raster images to describe the concepts that make up the model. This research is directed at developing a model that can be used for multimedia in any application, as opposed to much of the current research that has restricted the environment to some special application area in order to reduce the complexity to a manageable level. The rich semantics associated with multimedia data make it difficult to develop a model that will accommodate it. Since the relational model has been successfully used in dealing with formatted data in traditional DBMS applications, this project will try to use that success and extend it to cover multimedia data. [Ref. 4]

Since current technology does not provide a way for the DBMS to retrieve information from images, the user must provide this information. This is the basic philosophy behind this research. Information the image conveys is put into words, which then become the description of the image. The purpose behind all this is contents search, which is something that is not addressed by either ORION or MINOS. Once this is done, the traditional well-established and well-proven

techniques for information retrieval can be employed. This research project plans to implement a prototype using the Ingres Database Management System. [Ref. 4]

B. MULTIMEDIA REPRESENTATION FOR EXTENDED MODEL

The design breaks the representation of multimedia data into three parts.

1. Registration Data

This includes any data related to the physical aspects of the raw data for the device that will be used to display the raw data, such as its resolution and the color map for the image data. [Ref. 4]

2. Description Data

This is the user's description of the multimedia data [Ref.4]. For example, if the image was of a submarine heading out to sea from San Francisco, then the description would just use plain language and say something like the following.

USS Ohio departing San Francisco;

Golden Gate Bridge ahead;

Sea swells 5-10 feet;

Submarine surfaced.

Obviously, the description is extremely important because this is what will be used to actually search for multimedia data.

3. Raw Data

This is the actual bit string for the data [Ref. 4]. These three types of data are not new in themselves. The new thing here is the

textual description. It is this description that provides the ability to search for an image by searching for its textual description vice trying to actually search through the actual images.

C. MULTIMEDIA DATABASE MANAGEMENT

Multimedia data is largely made up of unformatted data. Trying to search for something would involve unrealistic search time if it were necessary to access these large data files. This problem is dealt with by introducing the description data. This description data allows the search to take place without ever having to actually access any images until the one being searched for is found. This allows for a reduction in complexity and allows the use of already well-proven techniques. Searching in raw data vs. searching in description data makes a difference not only in performance but also in quality. [Ref. 4]

The integration of the raw data and the registration data into the database system is accomplished by representing them as attribute values. This implementation follows the same principles as the Aggregate Data Manager (ADM) model, which is based on System R and uses SQL as its starting point. Further development in this area is planned for some future time. [Ref. 4]

Figure 6 shows an instance of the Abstract Data Type Image. The image is an attribute of a given object, and the registration data, description data, and raw data are internal to the Image attribute. The components of an Image value are made accessible through functions.

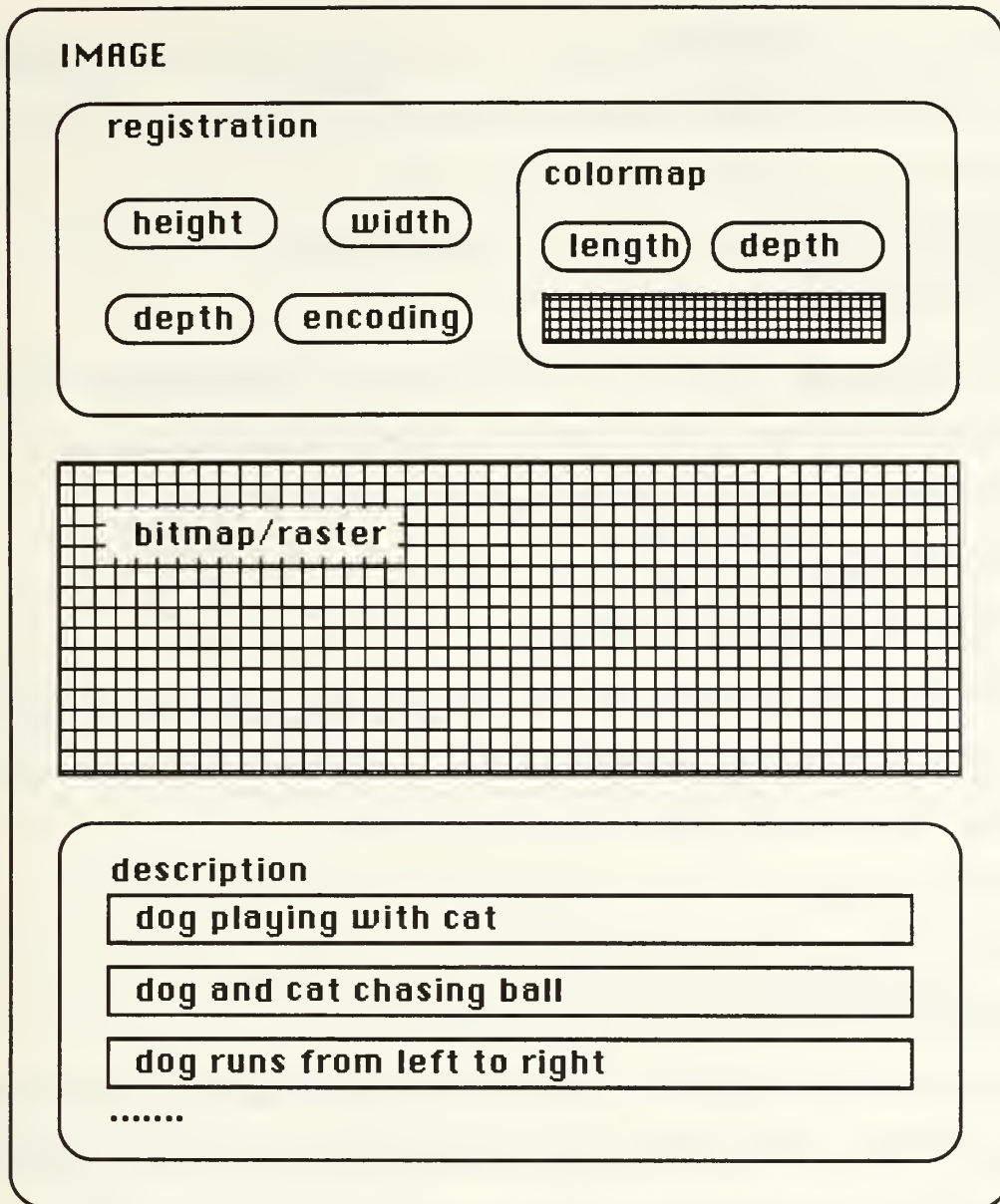


Figure 6. **Abstract Data Type Image**

D. THREE RELATIONAL SCHEMA TYPES FOR STORING IMAGES

An Image is treated here as an attribute of an object. It should be emphasized that the Images are attributes and not objects. There are three schema types identified for the storage of these Images

attributes, as identified in Figure 7. The first one is the simplest and is best used when there is only one image per object and each image contains only one object. The object will be something like a ship or an aircraft and it will be followed by a list of attributes. [Ref. 4]

OBJECT(O-ID ,..., O-IMAGE)

In some other applications, the number of images that belong to each object will vary. So in order to meet the requirements of first normal form, the second schema type is needed. [Ref. 4]

OBJECT(O-ID ,...)

OBJECT-IMAGE(O-ID , O-IMAGE)

The first two schema do not address the redundancy problem involved in storing the same image for a number of different objects. The third schema type addresses this problem.

OBJECT(O-ID ,...)

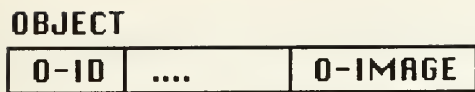
IMAGE-OBJECT(I-ID , I-IMAGE)

IS-SHOWN-ON(O-ID , I-ID , COORDINATES)

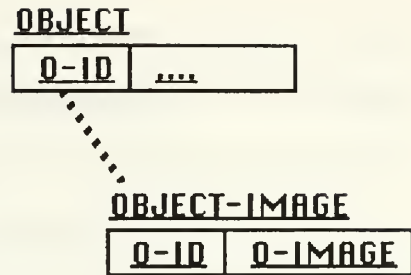
The coordinates provide the location of the given object on the image. The application will be the determining factor in deciding which of the three schemata should be used. [Ref. 4]

E. FUTURE RESEARCH

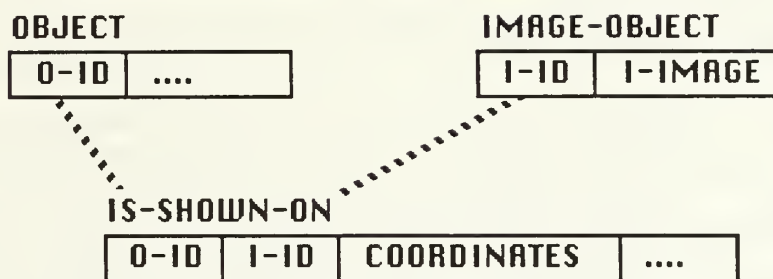
Work is currently in progress by the authors and one thesis student at the Naval Postgraduate School in Monterey to implement a simple prototype to handle only the image portion of multimedia type



a) Type 1



b) Type 2



c) Type 3

Figure 7. **Relational Schema Types for Storing Images**

data. A way to handle sound is also being investigated by another thesis student at the Naval Postgraduate School. Potential areas for future research can be found in investigating the integration of this data model and its ability to handle additional multimedia data types and their access functions. [Ref. 4]

F. ANALYSIS

Some of the more obvious weaknesses associated with this model are its built-in redundancy because the description only reiterates

what the image already contains. It is left in the user's hands as to whether the text description is any good or is instead ambiguous or inaccurate. Although redundancy is definitely a weakness in this model, it is also a weakness in all the systems outlined in this paper and so not exclusive to this model alone. All of the problems of record-oriented models described in Chapter II are still present in this model. The strongest disadvantage associated with this model is its lack of generalization.

This model does offer several very strong points. For one, there are two ways to represent descriptions. One is to use the description data and the other is by using the power of the relational model and defining a relation where each tuple represents an object and all of its attributes. For example, the image of a ship could be represented by a single tuple. A major advantage to this model is that it is easy to understand. This is a result of the familiarity and extensive prior use of the relational model.

The relational model does not, in theory, provide greater efficiency than the object-oriented model. It is only that at the current level of development the relational model's efficiency has been optimized to a greater degree than the object-oriented model. The extended relational model and its follow-on work are seen as being valuable as an intermediate step in the realization of a Multimedia Database Management System, while research continues toward the production of more efficient object-oriented systems.

VI. THE THREE MODELS COMPARED

A. INTRODUCTION

The purpose of this chapter is to take a closer look at the three models in terms of how different applications would be realized with each of the three systems. Two possible applications have been selected and common queries for each one will be used to show the objects/relations that would be required to produce a response in each model.

Certain applications will tend to favor one model over another. In order to reduce this, the applications selected are quite different so as to allow each model to highlight both its strengths and weaknesses.

B. APPLICATION ONE

The first scenario is a potential military or civilian application where satellite photos are saved and used later for briefings or reports. The most common requirement from the MDBMS would be the storage and retrieval of these photos. A query to retrieve the satellite photo image is seen as one of the most common queries for this application. Other possible queries for this application might be:

1. Retrieve satellite photo showing California at 0900 on 9 Nov 1987.
2. Retrieve all satellite photos showing drought regions.
3. Retrieve all satellite photos showing Soviet ship-building facilities.

In addition to the actual satellite photo image itself, other information will also be necessary. This information could be one or more of the following.

1. Position of the satellite.
2. Date and time associated with a given position.
3. Geographic coordinates of area on the photo.
4. Type of photographs (infrared, type of filter used, etc.).
5. Classification of photo.

The three different models need to be looked at in terms of how all this data is organized. This will mean in terms of relations, attributes, objects, et cetera. The criteria that will be used in the retrieval of the satellite photos needs to be determined. For instance, is a retrieval based on the coordinates of the area on the photo, or the type of photo, or perhaps the name of a particular geographic object?

1. ORION Model

For the ORION model, we need to determine what the objects will be and what operations will be performed on them. In this application, the Satellite-Photo will be an object. It will inherit methods from the Photo class since it will be a subclass of the Photo superclass. In addition to the methods it inherits, it will have some specific methods and attributes of its own, as shown in Figure 8.

As mentioned earlier, a common query for this application would be the retrieval of a particular satellite image. The message protocol for the presentation of multimedia information is described in

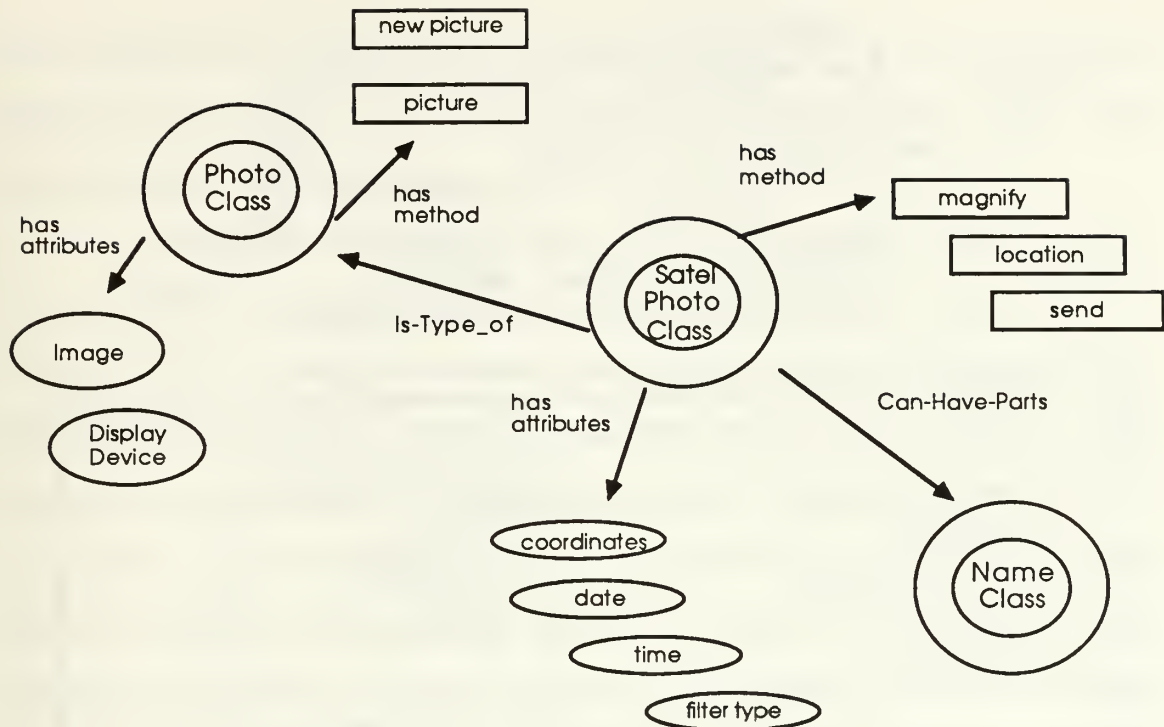


Figure 8. **Partial Photo Hierarchy**

the research done by Woelk, et al. Using the example given in their paper as a model, Figure 9 shows what happens when the Satellite-Photo instance receives the picture message. The picture method defined for the Satellite-Photo class will send a present message like the one below to whatever image presentation device is specified. [Ref. 11]

(present presentation-device captured-object[physical-resource])

If one satellite photo contains an image that includes several different objects like a River, a Country, and a Ship, this model can handle this situation very well. The objects for River, Country, and

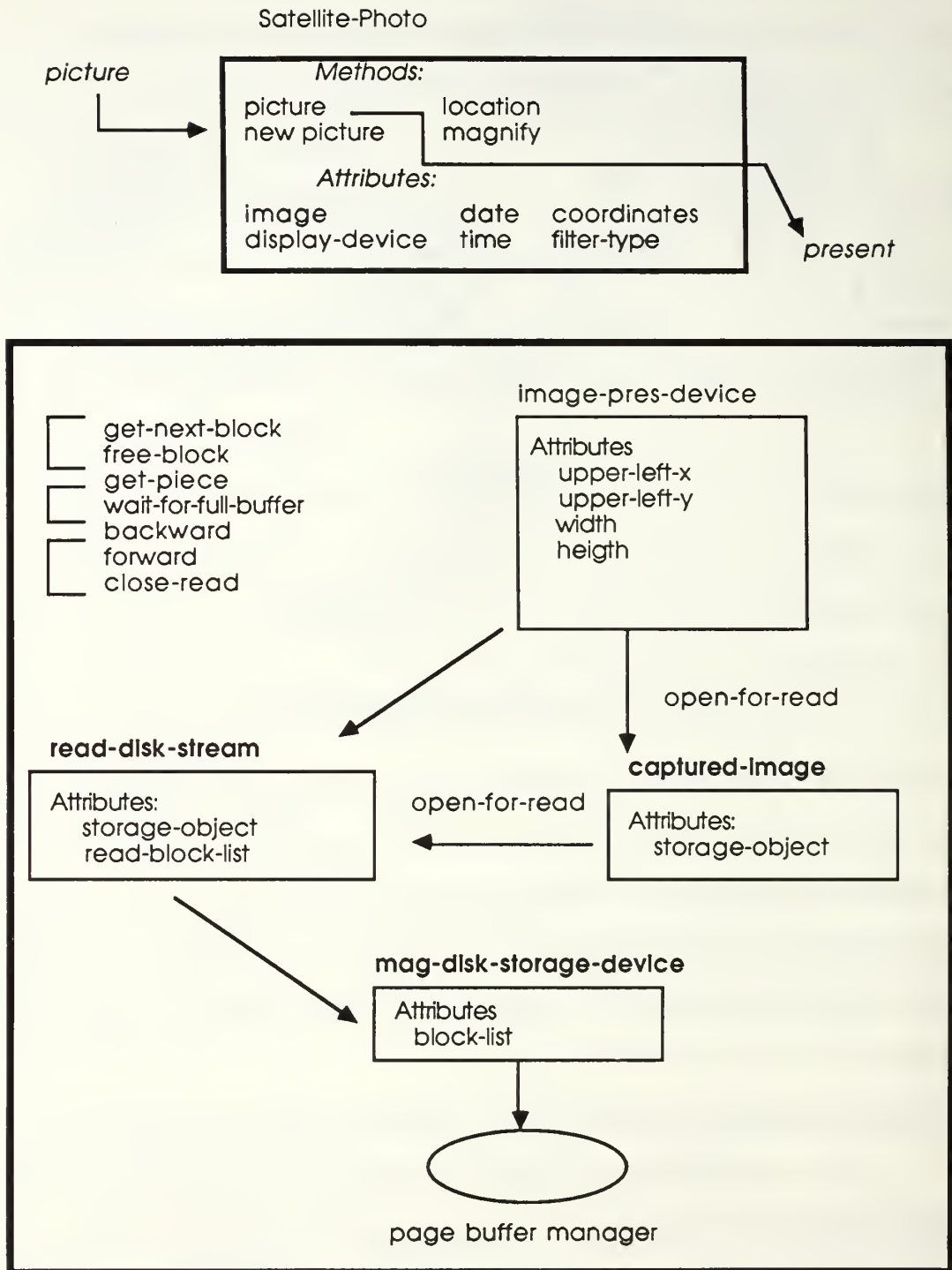


Figure 9. **Message-Passing Protocol**

Ship can all be related to the satellite photo via a Can-Have-Parts directed arc between the Satellite-Photo class and the River class, Country class, and Ship class.

Although not specifically addressed in the ORION papers researched, the system must be able to handle the user's need to interact with it through a browse mode and some form of query capability. The search for a particular satellite image might be based on its name, its coordinates, or some other feature that is defined for it. A text search could be made on text associated with the photo or some sort of textual description of the photo image. The browsing mode might be used when a query cannot be formulated that will go directly to the required information. A combination of these two modes should fulfill the users needs.

2. MINOS Model

This model was designed to support multimedia documents on a workstation equipped with image and audio inputs. Active multimedia document presentation and browsing within documents is seen as the most frequent requirement by these systems designers and so that is where they have placed their design emphasis. With the help of a menu and some graphics capabilities, the user can interactively specify his queries. He can call up a document or a part of a document for viewing or for use in the creation of a new document. If the user knows the name of the document that contains the satellite photograph he wishes to view, he can use the browse capability within that

document to search until he locates the photo in which he is interested.

Content addressability is realized in this model by letting the user specify queries on the attribute values, the text part, the image part, the voice part, or the presentation form of the documents. The queries on the attribute or text part are much like those handled by normal database management. Queries on the voice part can specify whether the document even has a voice part. Queries on the presentation form might refer to the size or location of the image on a page. The image part as per our application can reference text, which means referencing captions, text appearing within the image, or any related paragraphs. For example, a query could be made by searching for a certain location, assuming that all satellite photos have the location included as a part of the photo. If we want to find photos of Hurricane Gilbert, we can search for "Hurricane Gilbert" because it is likely that this will be a caption or associated text in any document containing photos of this specific hurricane.

In the MINOS model you expect to request multimedia information that lends itself to a document type of format. This is not always going to be the case with the types of applications that are needed. As pointed out earlier, the browsing feature is very strongly emphasized in this model. The question here is, do we really need browsing for the application we are looking at? It is very unlikely that a user will be interested in having to browse through satellite photos. This model was designed specifically with an office environment in

mind and does not really lend itself very well to this type of application. The satellite photo could be thought of as a part of a document or, in fact, the whole document itself. More likely, there will at least be the need for some textual information along with the image if these photos are going to be used as part of a report.

Images are represented in this model as objects. These objects can be organized into classes; the class hierarchy can have several levels. In this case, our satellite photo might have subclasses of images that are infrared photos and classified satellite photos. As shown in Figure 4, an Image Type can be defined which identifies the class. This class information can be used to help in identifying objects that contain a certain type of image. The presentation of the image is accomplished using the methods for that specific class of images, as shown in Figure 4. In order to extract information from the Image, the application will need to define the methods and interfaces that will allow it. [Ref. 2]

If the user finds the image he wants from a particular document, he must then move from the Browsing interface to the Extraction interface. Menu options are then provided for the extraction of various objects from the document. One of these menu options is for Image extraction.

3. Extended Relational Model

This model has been designed with a focus on making easier contents-oriented searches on images. This is accomplished through text descriptions that allow the user to describe the contents of the

image. The already well-known methods for text search and retrieval can then be used to find these textual descriptions. This model seems the most suited of the three for the storage and retrieval of stand-alone images.

In this model, the image is usually an attribute of some entity like Country, State, or River. In this case, we will look at the entity called Country. The way to assign the image to an object is to use one of the three relational schema shown in Figure 7. It is most likely that the number of images for the entity called Country will vary. If first normal form is required, this would mean using relational schema type 2, which would look something like the following.

COUNTRY(O-ID,)

OBJECT-IMAGE(O-ID, O-IMAGE)

Because there may be several images of the Country entity, O-IMAGE must be included with O-ID to make the key unique.

If the image shows several different Countries, then the image would need to be repeated in the relation for each of those Countries.. This redundancy could be very expensive. Relational schema type three is designed to avoid this redundancy. For our application schema, type 3 would look like the following.

COUNTRY(O-ID,)

IMAGE-OBJECT(I-ID, I-IMAGE)

IS-SHOWN-ON(O-ID,I-ID, COORDINATES,)

Obviously schema type 1 would be the least complicated, but the choice depends on the application. In our application, there will

probably be more than one country for each image, so schema type 3 will be necessary.

The value of the Image of our satellite photo would be represented much like Figure 6. In this application, the description might read something like the following:

description

Hurricane Gilbert 100 miles off coast of Texas;

Hurricane moving 5 mph in a northwesterly direction;

Current maximum winds are 105 mph;

Because this description is actually tied to the Image itself, it can be used to search for the Image. An example of how the description for a satellite photo of Hurricane Gilbert can be used to retrieve the Image might be as follows:

```
SELECT GET_RESOLUTION (I-IMAGE),....  
INTO $resolution,...  
FROM IMAGE-OBJECT  
WHERE CONTAINS (I-IMAGE,  
  "Hurricane | Gilbert*",  
  "Hurricane&moving*&northwesterly",  
  "maximum&current*");
```

The | symbol means that two words appear in the same sentence, the & symbol means that there may be other words between the two words specified, and the * symbol matches up strings of arbitrary length [Ref. 4].

C. APPLICATION TWO

This second scenario is much more complex than the first one and attempts to look even further into the future at what might be a very useful application of a multimedia database management system. This application would be designed for use by the military. It would involve the collection and integration of a variety of data types like radio messages, sonar, radar, intelligence reports, satellite photos, historical information, catalogues of ship types, reconnaissance photos, and maps.

Ideally, we would want the MDBMS to be robust enough to handle all these different types of data in a real-time environment like the combat information center on board a naval combatant. Advances in artificial intelligence that could take the information from all these sources and apply some reasoning in order to provide the officer in charge with timely decision alternatives would be the ultimate goal. Obviously, we are a long way from a system that can provide these capabilities, but we will look at how each of the three models might attempt to approach this type of application without considering the inability of any current system to provide responses in the time-critical fashion required by many potential applications.

Just as with application one, each of these types of data will need to have other information associated with it. For instance, the radio message would be an audio input and might need associated information like the following.

1. Frequency on which the signal is being received.

2. Classification of the message.
3. Date and time of the message.
4. Where the message is originating from.

Common queries for a system that is this complicated would not be simple retrieval and storage oriented. If the system provides the ability to integrate all these types of data, you would want it to respond to queries like the following.

1. What is the optimum current course adjustment?
2. Does the current situation warrant arming the weapon system?
3. What are the non-friendly ships within a 200-mile radius of the ship?

In order for these questions to be answered, each will have to be broken down into more detailed questions. For example, in looking at the first query, which asks for the optimum course adjustment, a number of questions will need to be answered in order to formulate a good response to the original query. Some of these questions might be:

1. What is the current position of hostile forces?
2. What is the current course of hostile forces?
3. What are the weapons systems of these hostile forces?
4. What is the current position of friendly forces?
5. What is the current course of friendly forces?
6. What are the weapons systems of these friendly forces?
7. What is the current status of stores on board?
8. What is the equipment casualty status?
9. Would the recommended course adjustment conflict with the current mission?

Once the original query is answered, the officer in charge will most likely want to personally investigate some of the input that went into making up that answer. He may want to view the reconnaissance photos or listen to the latest radio messages that apply. There must be some means made available for him to select them. Perhaps the system could prioritize the criteria that was used in making up the final answer. Then the officer in charge could look at the most critical items without having to look through all the criteria.

We could introduce the concept of a threat factor. It would be the function of the system to constantly update the threat factor value by evaluating the multimedia data that is available to it. This data will include those data types outlined earlier as a minimum. On the basis of the threat value, the system will either remain passive and wait for the user to query it or it will become active and provide information to the user without being queried. The system described now becomes the integration of multimedia database technology and artificial intelligence.

1. ORION Model

First it is necessary to decide what objects will be necessary and exactly what operations will need to be performed on those objects for this application. Just looking at one small portion of the necessary objects would define one for "Reconnaissance-Photos" and one for some sort of catalogue of "Non-Friendly-Ships." Figure 10 shows how these two objects might be represented. The Photo attribute contains a set of ID number values. The dotted lines show

that there is a relationship based on these values between the Photo attribute of the Ship object and the Reference-Photo object and Partial-Image object. These two were chosen for expanding on due to the obvious relationship between them.

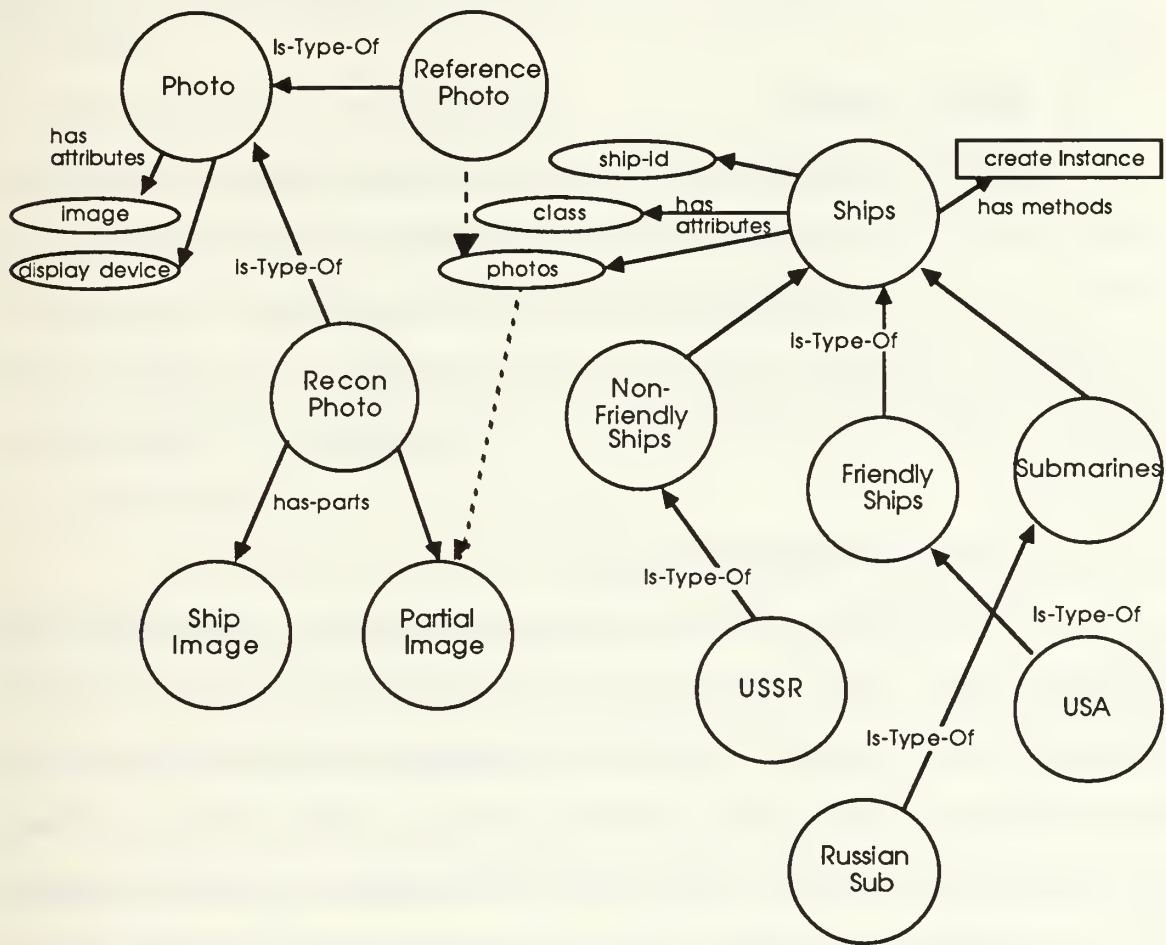


Figure 10. Representation of Reconnaissance Photo and Ships

The third query that was mentioned earlier will need to use information from both of these objects in addition to others to formulate a response. The reconnaissance photos that fall within the 200-

mile radius of the ships current location must first be located. This can be done based on the coordinates where each photo was taken and the ship's current location. Each Reconnaissance-Photo will need to be inspected to see whether it shows any ships. If it does, the catalogue for Non-Friendly-Ships will need to be compared for a possible match.

2. MINOS Model

With the emphasis on documents in this model, it is not well suited for all the requirements of this application. The menu-driven communication between the system and the user does not allow for the complex queries necessary for this application. This model is not generic enough to lend itself to a discussion of this specific application.

3. Extended Relational Model

This model has shown how it would deal with images, but this application goes much further with its requirements, covering the full spectrum of multimedia data types. It seems reasonable to think that new attribute types could also be used to deal with the other unformatted types of data, like sound, just as with the image data type. The contents-oriented search on audio data could just as easily be handled by text descriptions that describe the contents of the audio data.

Still looking at the reconnaissance photos and the catalogue of non-friendly ships as an example for this application, we would find the reconnaissance photos to be represented much like the example

given for application one. The catalogue of non-friendly ships could be represented using a relation something like this.

NON_FRIENDLY_SHIPS(country, missile-type, missile-qty, max-depth)

Some of the other possible relations for this application might be the following.

PHOTO(photo-id, photo-image)

REFERENCE-PHOTO(photo-id, ship-shown-id, data, location)

RECON-PHOTO(photo-id, data, location, camera)

SHIP-SHOWN(photo-id, ship-id, partial-image)

SHIPS(ship-id, class, ref-photo-id, country, max-speed, current-position, current-speed)

SUBMARINE(ship-id, max-depth)

This model relies on the user to provide the descriptions of the unformatted data for search purposes. It is obvious that this would not be acceptable for this application, but until the technology becomes available that will allow for a more complete interpretation of the data by the system and a more efficient method of search, it is one way of realizing a system that will provide the groundwork for the more advanced system we have described in this application.

VII. SUMMARY AND CONCLUSIONS

A. AREAS OF FURTHER RESEARCH

This paper has examined three different approaches to modeling a multimedia database system. Each of the three has both costs and benefits associated with the design decisions made by its authors. The attention paid to this area of research will continue to increase as technology advances and as the need for this type of data management becomes more and more apparent.

Possible follow-on work to this paper could involve the actual design of a new multimedia database system model. This model could involve a totally different approach from the three examined in this paper, or it could combine the positive qualities of these several models into one model, or it could be an extension to an already developed model. Partial implementation of a new or existing model also remains as a possible area for future research.

B. CONCLUSIONS

There are several conclusions that can be drawn from the discussion involved in this paper. First, we have seen the problems and confusions that arise from the lack of conformity in the definition and use of a variety of database terms and concepts. This confusion has made it very difficult to define the requirements of a multimedia database management system and just exactly what it is expected to accomplish. Second, the systems that are currently being developed are

usually aimed at a narrow application window in order to realistically provide some functionality. The current state of technical development is not yet advanced to a level that allows for a broader approach and, as stated earlier, there is as yet no consensus on just exactly what should be provided to make this possible. Finally, the awareness of the many areas where a multimedia database management system would be a valuable asset is increasing with time. It is the awareness of this need that will continue to drive the research in this area.

LIST OF REFERENCES

1. *Interactive Multimedia*, eds. Sueann Ambron and Kristina Hooper, pp. 4-6, Microsoft Press, 1988.
2. Christodoulakis, S., Theodoridou, M., Ho, F., Papa, M., and Pathria, A., "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System," *ACM Transactions on Office Information Systems*, vol. 4, no. 4, pp. 345-383, October 1986.
3. Woelk, D., Kim, W., and Luther, W., "An Object-Oriented Approach to Multimedia Databases," in Proc. ACM SIGMOD '86 Int. Conf. on Management of Data (Washington, D.C., May 1986), ed. C. Zaniolo, *ACM SIGMOD Record*, vol. 15, no. 2 (June 1986), pp. 311-325.
4. Naval Postgraduate School Report No. NPS52-88-024, *Image Database Management in a Multimedia System*, by K. Meyer-Wegener, V. Lum, and C. T. Wu, Monterey, CA, August 1988.
5. Woelk, D., Luther, W., and Kim, W., "Multimedia Applications and Database Requirements," Proceedings of the IEEE CS Office Automation Symposium (Gaithersburg, MD) pp. 180-189, April 1987.
6. Rishe, N., *Database Design Fundamentals*, p. 4, Prentice-Hall, 1988.
7. Naval Postgraduate School Report No. NPS52-87-050, *Integrating Advanced Techniques into Multimedia DBMS*, by V. Y. Lum, C. T. Wu, and D. K. Hsiao, Monterey, CA, November 1987.
8. Kent, W., "Limitations of Record-Based Information Models", *ACM Transactions on Database Systems*, vol. 4, no. 1 (March 1979), pp. 107-131.
9. Nierstrasz, O. M., "A Survey of Object-Oriented Concepts," *Active Object Environments*, ed. D. Tsichritzis, pp. 1-17, Centre Universitaire D'informatique, Universite De Geneve, Switzerland, 1988.
10. Korth, H., and Silberschatz, A., *Database System Concepts*, pp. 40-44, McGraw-Hill, Inc., 1986.

11. Woelk, D., and Kim, W., "Multimedia Information Management In An Object-Oriented Database System," Proceedings of the 13th VLDB Conference, Brighton, England, 1987.

BIBLIOGRAPHY

Booch, G., *Software Engineering With Ada*, The Benjamin/Cummings Publishing Company, Inc., Menlo Park, CA, 1986.

Christodoulakis, S., Ho, F., and Theodoridou, M., "The Multimedia Object Presentation Manager of MINOS: A Symmetric Approach," in Proc. ACM SIGMOD '86 Int. Conf. on Management of Data (Washington, D.C., May 1986), ed. C. Zaniolo, *ACM SIGMOD Record*, vol. 15, no. 2 (June 1986).

Christodoulakis, S., Vanderbroeck, J., Li, J., Li, T., Wan, S., Wang, Y., Papa, M., and Bertino, E., "Development of a Multimedia Information System for an Office Environment," Proceedings Very Large Databases Tenth International Conference on Very Large Databases, Singapore, August 1984.

Duff, C., "Designing An Efficient Language," *Byte Magazine*, August 1986.

Masunaga, Yoshifumi, "Multimedia Databases: A Formal Framework," IEEE Computer Society Office Automation Symposium, 27-29 April 1987, Gaithersburg, MD.

Naval Postgraduate School Report No. NPS52-88-010, "Managing Multimedia Data—An Exploration by K. Meyer-Wegener, V. Y. Lum, and C. T. Wu, Monterey, CA, March 1988.

Naval Postgraduate School Report No. NPS52-88-025, "A Conceptual Design of a Multimedia DBMS for Advanced Applications," by V. Lum and K. Meyer-Wegener, Monterey, CA, August 1988.

Naval Postgraduate School Report No. NPS52-88-047, "Multimedia Databases: Paradigm, Architecture, Survey and Issues," by P. C. Lockemann, Monterey, CA, September 1988.

Sato, M., Koda, M., Ioka, M., and Hong, J., "Image/Text Retrieval System on a Lan," IEEE Computer Society Office Automation Symposium, 27-29 April 1987, Gaithersburg, MD.

Sventek, J., "An Architecture Supporting Multi-Media Integration," IEEE Computer Society Office Automation Symposium, 27-29 April 1987, Gaithersburg, MD.

Naval Postgraduate School Report No. NPS52-87-030, *GLAD: Graphics Language for Database*, by C. T. Wu, Monterey, CA, July 1987.

The Xerox Learning Research Group, "The Smalltalk-80 System," *Byte Magazine*, August 1981.

INITIAL DISTRIBUTION LIST

	<u>No. Copies</u>
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	1
4. Curriculum Officer, Code 37 Computer Technology Naval Postgraduate School Monterey, CA 93943-5000	1
5. Superintendent, Naval Postgraduate School Computer Technology Programs, Code 37 Naval Postgraduate School Monterey, CA 93943-5000	1
6. Professor C. Thomas Wu, Code 52Wq Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	2
7. Klaus Meyer-Wegener, Code 52Mw Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	1
8. Lt. Diane M. Enbody 4693 Blue Pine Circle Lake Worth, FL 33463	4

Thesis

E43655 Enbody

c.1 Analysis of existing
advanced data models and
their applicability as a
model for a multimedia
database management
system.

23 AUG 90

36209

25 MAY 93

39018

31 JUL 93

38545

Thesis

E43655 Enbody

c.1 Analysis of existing
advanced data models and
their applicability as a
model for a multimedia
database management
system.



thesE43655

Analysis of existing advanced data model



3 2768 000 81230 9

DUDLEY KNOX LIBRARY